(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 368 144 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
07.02.1996 Bulletin 1996/06

(51) Int Cl.[6]: **G06F 1/32**

(21) Application number: **89120284.8**

(22) Date of filing: **02.11.1989**

(54) Digital computing system with low power mode

Digitalrechnersystem mit Niederstromverbrauchmodus

Système de traitement numérique avec un mode faible consommation d'énergie

(84) Designated Contracting States:
DE FR GB IT NL

(30) Priority: **10.11.1988 US 269344**

(43) Date of publication of application:
16.05.1990 Bulletin 1990/20

(73) Proprietor: **MOTOROLA, INC.**
Schaumburg, IL 60196 (US)

(72) Inventors:
- **Fourcroy, Antone Louis**
  Austin, Texas 78750 (US)
- **McDermott, Mark William**
  Austin, Texas 78759 (US)
- **Dunn, John P.**
  Austin, Texas 78737 (US)
- **Burgess, Bradley G.**
  Austin, Texas 78749 (US)

(74) Representative: **Dunlop, Hugh Christopher et al**
Basingstoke, Hampshire RG21 7PL (GB)

(56) References cited:
EP-A- 0 050 844
US-A- 4 748 559

EP 0 368 144 B1

Printed by Jouve, 75001 PARIS (FR)

Description

Field of the Invention

The present invention relates, in general, to a digital computing system having a low power mode. More particularly, the invention relates to a digital computing system which prepares to enter a low power mode by broadcasting status information relating to the conditions for exit from the low power mode.

Background of the Invention

Digital computing systems, in particular integrated circuit computing systems, commonly have the ability to enter a low power mode during which time no processing is accomplished and various subsystems are shut down to reduce power consumption. It is common to provide that the occurrence of predetermined external events will "awaken" the system from its low power mode to resume normal processing.

U.S. Patents 4,748,559 and 4,758,945, both assigned to the assignee of the present invention, disclose aspects of a digital computing system which responds to particular software instructions by entering one of two available low power modes. The described system is available as an integrated circuit designated the MC146805 from Motorola, Inc. of Austin, Texas. Either of the two available low power modes disclosed can be terminated by a reset or interrupt event. In the case of interrupt events the mask bit, which prevents certain interrupts from being recognized by the system, must be cleared in order that a maskable interrupt event will terminate the low power mode.

The above-mentioned patents describe a system which is well suited to traditional "hand-packed" integrated circuit design methodology. However, as integrated circuit computing systems move toward "modular" design methodologies in order to permit more rapid design of customized systems, some reset and interrupt control circuits may be removed from close logical and physical proximity to the central processing unit. In this case, the previously known techniques for terminating a low power mode require modification.

European Patent Application 0 050 844 describes a method of controlling a clock signal supplied to a logic circuit in a data processing apparatus so as to inhibit the supply of the clock signal to the logic circuit or to fix the level of the clock signal at a specific level in order to reduce power consumption. This patent application also teaches a clock signal supply inhibit instruction which, when read out, inhibits the supply of the clock signal to the logic circuit or fixes the level of the clock signal to a specific signal level. The clock signal is again supplied to the logic circuit in response to the application of an interrupt signal."

Summary of the Invention

In accordance with a first aspect of the invention there is provided a digital computing system according to claim 1.

In accordance with a second aspect of the invention there is provided a method for executing an instruction in a digital computing system according to claim 7.

Brief Description of the Drawings

One digital computing system utilizing the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 is a block diagram of an integrated circuit digital computing system according to a particular embodiment of the present invention;

FIG. 2 is a block diagram of the central processing unit of the computing system of FIG. 1;

FIG. 3 illustrates the register set of the central processing unit of FIG. 2; and

FIG. 4 is a timing diagram illustrating several bus cycles as executed by the central processing unit of FIG. 2.

Detailed Description of the Invention

The terms "assert", "assertion", "negate" and "negation" will be used to avoid confusion when dealing with a mixture of "active high" and "active low" signals. "Assert" and "assertion" are used to indicate that a signal is rendered active, or logically true. "Negate" and "negation" are used to indicate that a signal is rendered inactive, or logically false. In addition, the terms "set" and "clear" will be used when referring to the rendering of a status bit or similar apparatus into its logically true or logically false state, respectively.

FIG. 1 illustrates an integrated circuit computing system according to a particular embodiment of the present invention. A microcomputer 10 comprises a central processing unit (CPU) 11, an inter-module bus (IMB) 12, a serial commu-

nication interface 13, on-board memory 14, a timer module 15 and a system integration module (SIM) 16. Inter-module bus 12, which comprises multiple data, address and control signal lines as described in detail below, is coupled to and provides for communication between each of the other components of microcomputer 10. Serial interface 13 provides for synchronous and/or asynchronous serial data transfer between microcomputer 10 and external devices and systems by means of several serial I/O pins. Memory 14 provides storage space for software instructions and other data useful to microcomputer 10. Timer module 15 provides various timing functions such as input capture, output compare and the like by means of several timer pins and is coupled to memory 14 by means of an interface 17. SIM 16 provides an interface between IMB 12 and an external bus, the details of which are discussed below, and also provides certain system functions such as clock signal generation and distribution

The tables which appear immediately below contain the signal definitions for all of the lines of IMB 12 and the external bus which is coupled to SIM 16. Both of these buses are parallel communication buses.

TABLE I

INTER-MODULE BUS SIGNALS

SIGNAL NAME	MNE-MONIC	FUNCTION	DIRECTION
Address Bus	ADDR0-ADDR23	24 bit address bus	output
Data Bus	DATA0-DATA15	16 bit data bus capable of 8 and 16 bit transfers	input/output
Function Code	FC0-FC2	Identifies CPU state (supervisor/user) and address space of the current bus cycle	output
Clock	CLOCK	Master system clock	input
Cycle Start	CYS	Indicates start of internal bus cycle	output
Address Strobe	AS	Indicates second phase of bus cycle and that address is valid	output
Data Strobe	DS	Indicates third phase of bus cycle and that data is valid	output
Read/Write	WRITE	Defines a bus cycle as a read or write, relative to the bus master	output
Transfer Size	SIZ0-SIZ1	Specifies the number of bytes yet to be transferred in a bus cycle	output
Data Transfer Acknowledge	DTACK	Slave response which terminates a bus cycle	input
Bus Error	BERR	Provides for termination of a bus cycle if no valid response is received	input
Relinquish and Retry	RRT	Provides a means for breaking a bus mastership standoff at the internal/external bus boundary	input
Retry	RETRY	Provides for termination of a bus cycle which should be rerun	input
Halt	HALT	Indicates that the CPU has halted due to an abnormal condition	output
Breakpoint Request	BKPT	Signals a request for a breakpoint on the current bus cycle	input
Breakpoint Acknowledge	FREEZE	Indicates that the CPU has entered background debug mode	output

TABLE I (CONT'D)

System Reset	SYSRST	Provides a "soft" reset which does not disturb system configuration data	output
SIGNAL NAME	MNE-MONIC	FUNCTION	DIRECTION
Master Reset	MSTRST	Provides a "hard" reset of everything	input
Interrupt Request Level	IRQ1 - IRQ7	Prioritized interrupt requests to the CPU	input
Autovector	AVEC	Specifies that autovector feature is to be used during an interrupt ack. cycle	input
Bus Request	BR0 - BRn	Prioritized bus mastership arbitration signals	input
Bus Lock	BLOCK	Allows bus master to retain bus	output
Test Mode	TSTMOD	Enables test mode for all devices	input
Enable IMB test lines	IMBTEST	Changes function of IRQ1-IRQ7 to test lines	input

Note that signal directions in the table above are specified with respect to CPU 11.

TABLE II

EXTERNAL BUS SIGNALS

SIGNAL NAME	MNE-MONIC	FUNCTION	DIRECTION
Address Bus	A0-A23*	24 bit address bus	input/output
Data Bus	D0-D15	16 bit data bus capable of 8 and 16 bit transfers	input/output
Function Code	FC0-FC2*	Identifies CPU state (supervisor/user) and address space of current bus cycle	input/output
Boot Chip Select	CSBOOT	Programmable chip select for boot-up	output
Bus Request	BR*	Bus Mastership Request line	input/output
Bus Grant	BG*	Bus Mastership Grant Line	output
Bus Grant Acknowledge	BGACK*	Bus Mastership Grant Acknowledge line	input/output

TABLE II (CONT'D)

SIGNAL NAME	MNEMONIC	FUNCTION	DIRECTION
Data and Size Acknowledge	$\overline{DSACK0}$ - DSACK1	Indicates that data is valid on read and that data has been received on write. Also indicates port size.	input/output
Address Strobe	AS	Indicates that address, function codes, etc. are valid	output/input
Data Strobe	DS	Indicates that data is valid on write and that slave should drive data on read	input/output
Read/Write	WRITE	Defines a bus cycle as a read or write, relative to the bus master	input/output
Transfer Size	SIZ0-SIZ1	Indicates single or multi-byte transfer	input/output
Bus Error	BERR	Provides for termination of a bus cycle if no valid response is received	input/output
Halt	HALT	Indicates that the CPU has halted due to an abnormal condition	input/output
Interrupt Request Level	IRQ1 - IRQ7	Prioritized interrupt requests to the CPU	input
Autovector	AVEC	Specifies that autovector feature is to be used during an interrupt ack. cycle	input/output
Bus Lock	BLCK	Indicates indivisible bus cycles	input/output
Reset	RESET	System Reset	input/output
External System Clock	CLK	External system clock – bus clock	output
Crystal Osc.	EXTAL, XTAL	Pins for connection of external oscillator or clock circuit	input/output
External Filter Capacitor	XFC	Allows connection of an external filter capacitor to internal clock circuit	output
Synthesizer Power Supply	V_{DDSYN}	Provides power to internal clock synthesizer	input
Main Power Supply	V_{DD}	Provides power to chip	input
Freeze	FRZ/QUOT	Acknowledges entry into background mode and outputs quotient bits in test mode	output

Test Mode Enable and Tri-State Control	$\overline{\text{TSTME}}$/TSC	Enables test mode or causes output drivers to be tri-stated	input
Clock Mode Select	MODCK	Selects source of system clock	input

The pins denoted above with an asterisk, the address pins A19-A23, the function code pins FC0-FC2, the bus request pin BR, the bus grant pin BG and the bus grant acknowledge pin BGACK, may also be used as programmable chip select pins. This feature of microcomputer 10 is not related to an understanding of the present invention. Signal directions are specified with respect to microcomputer 10.

Among the functions of SIM 16 is that of determining when a bus cycle initiated by CPU 11 is directed to a device external to microcomputer 10. If this is the case, SIM 16 executes an appropriate bus cycle on the external bus and also mediates between the internal bus cycle and the external bus cyde. In addition, SIM 16 is capable of displaying internal bus cycles which are directed at modules internal to microcomputer 10 via the external bus. This feature is useful for debugging and development purposes, among others.

FIG. 2 illustrates, in very simplified form, the internal structure of CPU 11 of FIG. 1. Fundamentally, CPU 11 comprises a micro-machine 20, an execution unit 21, a set 22 of registers and a bus interface 23. Micro-machine 20 is coupled bi-directionally to interface 23 and to execution unit 21. Registers 22 and execution unit 21 are coupled to one another by means of internal buses and the like which are not illustrated here. Execution unit 21 is also bi-directionally coupled to interface 23. Interface 23 is coupled to the address, data and control signals which comprise IMB 12.

Micro-machine 20 is responsible for determining the sequence in which instructions are to be executed, receiving the instructions from interface 23 after they have been fetched from memory (either memory module 14 or external memory), instructing interface 23 to perform instruction fetches and operand read or write cycles and decoding instructions into a plurality of control signals for use in controlling execution unit 21. As a portion of the instruction sequencing function of micro-machine 20, it performs exception processing, including the function of determining whether to acknowledge interrupt requests received via interface 23 from IMB 12. Execution unit 22 is responsible for the actual execution of the logical, arithmetic and other functions encoded in the instructions received by micro-machine 20. Registers 22 store various inputs to and results of the operations of execution unit 21. IMB interface 23 is a master-only interface to IMB 12. That is, it can initiate read and write cycles of IMB 12, and it can permit another master to initiate such cycles, but it cannot respond to either a read or a write cycle of IMB 12 which is initiated by another bus master.

Referring now to FIG. 3, register 22 of FIG. 2 are illustrated in greater detail. Registers 22 comprise 8 32-bit data registers, designated D0-D7, 7 32-bit address registers, designated A0-A6, 2 stack pointers, designated USP (for user stack pointer) and SSP (for supervisor stack pointer), respectively, a single 32-bit program counter, designated PC, a single 16-bit status register, designated SR, 2 3-bit function code registers, designated SFC (for source function code) and DFC (for destination function code), respectively, and a single 32-bit vector base register, designated VBR. The two stack pointers are alternately referred to with the designations A7 and A7', respectively.

Together, registers 22 comprise what is referred to as the programmer's model of CPU 11. The programmer's model illustrated here will be familiar to any user of microprocessors of the 68000-family of microprocessors available from Motorola, Inc. of Austin Texas.

For purposes of the present invention, only bits 8-10 of the status register SR are particularly relevant. These bits, designated I0, I1 and I2, respectively, comprise an interrupt mask. These three bits, which can encode 8 different interrupt mask settings, participate in the implementation of a prioritized interrupt recognition scheme. Basically, any interrupt source, whether internal or external, must identify its current interrupt priority level setting to CPU 11 in connection with its assertion of an interrupt request. If a requesting interrupt source has a priority level setting higher than the current mask value encoded in bits 8-10 of the status register, then the interrupt will be recognized. If the priority value is less than or equal to (except in the case of level 7 interrupts) the mask value, then the interrupt will not be recognized. Table III, below, illustrates the interrupt mask encoding scheme.

TABLE III

INTERRUPT MASK ENCODING			
MASK BITS			MASKED INTERRUPT
I2	I1	I0	PRIORITY LEVELS
0	0	0	none

Continuation of the Table on the next page

TABLE III (continued)

INTERRUPT MASK ENCODING			
MASK BITS			MASKED INTERRUPT
I2	I1	I0	PRIORITY LEVELS
0	0	1	0-1
0	1	0	0-2
0	1	1	0-3
1	0	0	0-4
1	0	1	0-5
1	1	0	0-6
1	1	1	0-6

The priority level of an interrupt request is determined by the levels of the seven interrupt request lines $\overline{IRQ1}$ - $\overline{IRQ7}$. An interrupt source having a priority setting of 7 uses $\overline{IRQ7}$ to generate an interrupt request, a source with a setting of 6 uses $\overline{IRQ6}$ to do so, and so forth.

Under normal operating conditions, all interrupts requests are communicated to CPU 11. Interrupt logic internal to CPU 11 compares the priority level of each interrupt request to its current mask setting and, if appropriate, commences an exception processing sequence.

The normal read and write cycles of IMB 12 will be described with reference to FIG. 4, which is a timing diagram illustrating those cycles. The signals illustrated are those of IMB 12. The cycles defined for the external bus are fundamentally similar to those described. The basic internal read and write cycles of IMB 12 (*i.e.*, cycles directed to one of the internal modules of microcomputer 10) each take place over the span of two complete cydes of CLOCK. the master system dock signal. The four phases, or tics, of CLOCK which occur during the basic bus cycles are numbered 1-4 and correspond to four states of the bus cycles.

An internal read cycle begins with the assertion of $\overline{CYS}$ during state 4. The bus master also drives the address and function code lines at this time, as well as negating $\overline{WRITE}$. During this period, IMB 12 is precharging $\overline{AS}$, $\overline{AACK}$ and $\overline{DTACK}$.

During the next clock phase, state 1, the bus master asserts $\overline{AS}$ and the slave which is to respond to this cycle asserts $\overline{AACK}$. Also, IMB 12 precharges the data lines and $\overline{DS}$ during state 1.

At the beginning of state 2, the bus master asserts $\overline{DS}$. The slave may begin driving the data lines as early as the beginning of state 2.

The slave should respond to the bus cycle by asserting $\overline{DTACK}$ or an appropriate error signal during state 3. The master samples $\overline{DTACK}$ and the error signals at the end of state 3 and, if neither is asserted, the master inserts a wait state (referred to as 3*), then returns to state 3 to sample $\overline{DTACK}$ and the error signals again.

By the beginning of state 4, the slave must have begun driving the data lines and stops asserting $\overline{DTACK}$. This completes the basic internal read cyde.

The basic internal write cycle is very similar to the cycle described above except that $\overline{WRITE}$ is asserted in state 4 and that the bus master drives the data beginning in state 2. Otherwise, the write cycle is identical to the read cyde.

The basic external read and write cydes are fundamentally the same as the corresponding internal cycles but for the insertion of wait states (state 3*) in each cycle. This insertion is performed by SIM 16 to "hold off" the termination of the IMB cycle while the slower, external bus completes its cyde. The external bus has a five state basic cycle.

A low power mode is defined for microcomputer 10 which is initiated by the execution of a particular instruction, LPSTOP, by CPU 11. This instruction comprises three words (a total of 48 bits). The first two words contain the particular bits which identify the LPSTOP instruction (the opcode) and the third word contains immediate data. When the LPSTOP instruction is received and decoded by micro-machine 20 (FIG. 2), a number of control signals are produced which cause execution unit 21 and bus interface 23 to perform certain tasks. First, one or more control signals are produced which cause execution unit 21 to place the immediate data portion of the LPSTOP instruction into the status register SR. This has the effect of resetting the interrupt mask bits (along with the other control and condition code bits in the status register) to the values specified in the immediate data field. Next, one or more control signals are produced which cause the program counter to be incremented so as to point to the location of the next instruction to be fetched. Finally, one or more control signals are produced which cause bus interface 23 to execute a special bus cycle, the LPSTOP cyde.

The LPSTOP cycle is fundamentally a normal, internal write cycle as is described above. The LPSTOP cycle is identified as different from other write cycles by the values of the function code signals (FC0-FC2) and of certain of the

address signals (A16-A19).

The function code signals identify each read or write cycle initiated by CPU 11 as being addressed to one of several possible address spaces. The various address spaces and function code signal encodings are set forth in Table IV.

TABLE IV

FUNCTION CODE ASSIGNMENTS			
FC2	FC1	FC0	ADDRESS SPACE
0	0	0	UNDEFINED
0	0	1	USER DATA
0	1	0	USER PROGRAM
0	1	1	UNDEFINED
1	0	0	UNDEFINED
1	0	1	SUPERVISOR DATA
1	1	0	SUPERVISOR PROGRAM
1	1	1	CPU

For an LPSTOP cycle the function code signals are all equal to one, making it a CPU space cycle. There are several other CPU space cycles (breakpoint and interrupt acknowledge, for example), so address lines A16-A19 are used to distinguish the CPU space cycles from one another. For the LPSTOP cyde, A19 and A18 are equal to 0 and A16 and A17 are equal to 1.

The LPSTOP cyde is one example of a special register access cycle. All special register access cycles have the function code and A16-A19 encodings described above. The lower sixteen address signals specify which special register is being accessed. In the preferred embodiment, the only special register implemented is the interrupt mask register in SIM 16, which is the destination of the LPSTOP cycle. In the general case, address signals A12-A15 identify the chip, signals A8-A11 identify the module and A0-A7 identify the special register which is the target of the special register access cycle. In the preferred embodiment, signals A0-A15 are all equal to 1 for the LPSTOP cyde.

The lower three lines of the data bus (DATA0-DATA2) are used to communicate bits 8-10 of the status register (I0-I2) during the LPSTOP cycle. SIM 16 responds to the LPSTOP cycle by storing the interrupt mask bits in its interrupt mask register.

The LPSTOP cycle is intended primarily to notify internal modules that low power mode entry is imminent and to communicate the interrupt mask bits to SIM 16. However, it is possible that devices external to microcomputer 10 may also need to be notified of the coming low power mode. Therefore, if the external bus is not under the control of an external bus master when the LPSTOP cycle is executed, the LPSTOP cycle will be executed on the external bus by SIM 16, so that external devices can prepare, if necessary, for the low power mode.

In addition to storing the interrupt mask bits, SIM 16 responds to the LPSTOP cycle by halting the IMB clock signal, CLOCK. CPU 11 and all of the other internal modules of microcomputer 10 use CLOCK as the sole source of fundamental internal timing. Thus, when CLOCK is halted, all of these modules are also halted. This greatly reduces power consumption. SIM 16 continues to generate clock signals for its own use, and so remains "awake" during the low power mode. The externally-supplied dock signal, CLK, may or may not be halted during the low power mode, depending on the state of a control bit which is set within SIM 16 under control of CPU 11.

Events which can terminate the low power mode are resets (an external device holding the $\overline{\text{RESET}}$ pin low for a predetermined period of time) and interrupts which have a priority level sufficiently high not to be masked by the interrupt mask bits stored in the interrupt mask register of SIM 16. Since all of the internal modules other than SIM 16 are shut down during the low power mode, none of them can produce an interrupt signal to terminate the low power condition. However, since SIM 16 itself contains certain monitors (*e.g.*: watchdog timers, periodic interrupt circuits and the like) which are capable of generating interrupts, and since SIM 16 remains active during the low power mode, the interrupt which terminates the power down condition can arise within microcomputer 10. In the particular embodiment being described, of the interrupt sources within SIM 16, only the periodic interrupt circuit remains active during the low power mode. Of course, it is also possible that an external circuit will be the source of the interrupt.

During the low power mode of operation, SIM 16 simply awaits either a reset or interrupt event. Any reset event causes SIM 16 to resume the generation of the CLOCK signal and to assert the $\overline{\text{MSRST}}$ signal, thus resuming normal program execution. Any interrupt event which has a sufficiently high priority to exceed the level masked by the interrupt mask bits passed to SIM 16 by the LPSTOP cycle will also cause an exit from the low power mode. In the case of an interrupt, SIM 16 will resume generating CLOCK and also present the interrupt request to CPU 11 on the $\overline{\text{IRQ1}}$ - $\overline{\text{IRQ7}}$

lines of IMB 12. CPU 11 will respond to this interrupt request as it would to any other such request, will execute appropriate exception processing routines and will return to normal program execution with the instruction which follows the LPSTOP instruction which initiated the low power mode.

Other than during the power down condition which follows the execution of the LPSTOP instruction, the interrupt mask register within SIM 16 is ignored. All internally generated interrupts are passed directly to CPU 11 via IMB 12. CPU 11 performs the comparison necessary to a determination of whether the interrupt signal will be acknowledged. All externally generated interrupts are passed by SIM 16 directly from the external bus interrupt lines to those of IMB 12 unconditionally.

The separation of the low power mode termination logic from the normal reset and interrupt logic in the CPU permits the CPU to be completely shut down during the low power mode, thus saving considerable power. Allowing the System Integration Module to perform the mask level comparison independently permits low priority interrupts to be completely ignored, obviating the need to "awaken" the CPU to perform the comparison. The present invention is not limited to the situation in which an interrupt mask setting is the critical determinant of exit from the power down mode. It may be some other conditional evaluation which the central processing unit would normally make, but which is advantageously made elsewhere in the system during a low power mode of operation. In addition, the details of the LPSTOP cycle described are not critical to the functioning of the invention. Any means of communicating the conditions by which exit from the low power mode is qualified from the central processing unit to some portion of the system which will remain active during the power down condition may be substituted for the disclosed LPSTOP bus cycle.

While the present invention has been shown and described with reference to a particular embodiment thereof, those skilled in the art will recognize various modifications. For instance, while the invention is disclosed in the context of a microcomputer comprising certain modules, any of these modules might be replaced by other modules having different functionality. In addition, although the central processing unit of the preferred invention is a microcoded machine, the invention might readily be implemented in the context of a hard-wired machine. Furthermore, the particular embodiment described implements the cessation of clock signals by halting the production of those signals at their source. An alternate implementation would be to continue to produce and distribute a clock signal during the low power mode and also to produce an LPSTOP control signal which would be distributed to all internal modules. At each module, logic would respond to the LPSTOP control signal by either blocking the clock signals, thus shutting down the module, or not blocking the clock signals, thus allowing the module to continue operation during the low power mode. This alternate embodiment would entail increased power dissipation during the low power mode, but would provide increased flexibility by allowing some modules to continue operation during the low power mode.

Claims

1. A digital computing system (10) comprising an inter-module communication bus (12);

   a central processing unit (11) comprising:

   register means (22) for storing an interrupt mask;
   a bus controller (23) coupled to the inter-module communication bus (12);
   first interrupt means (20) for selectively responding to an interrupt request received via the bus controller (23) from the inter-module communication bus(12) dependent on a comparison between a priority level of the interrupt request and the interrupt mask; and
   execution means (21) for executing instructions; and

   an integration module (16) coupled to the inter-module communication bus (12) comprising:

   clock signal generation means (16) for providing, via the inter-module communication bus (12), a clock signal to the central processing unit;

   characterized in that:
   the execution means is responsive to a particular instruction by causing the bus controller (23) to place predetermined signals and the interrupt mask on the inter-module communication bus (12), and
   the clock signal generation means ceases to provide the clock signal to the central processing unit in response to the receipt of the predetermined signals and interrupt mask from the inter-module communication bus (12).

2. A digital computing system (10) according to claim 1 further comprising:

first logic means (16) coupled to the communication bus for storing the signals representing the interrupt mask:
second logic means (16) coupled to the first logic means and being active only if the clock signal generation logic is not providing the clock signal to the central processing unit for responding to an interrupt request signal by comparing a priority level of the interrupt request signal to the contents of the first logic means (16);
third logic means (16) for selectively causing the clock signal generation logic to resume providing the clock signal to the central processing unit dependent on a result of the comparison performed by the second logic means; and
the clock signal generation logic continuously providing a clock signal to the first, second and third logic means.

3. A digital computing system according to claim 1 or 2 further characterized in that

the execution means (21) further responds to the particular one of the instructions by storing in the register means (22) an immediate bit field of the particular one of the instructions.

4. A digital computing system (10) according to claim 1 further comprising:

interrupt mask means (16) for storing, in response to the receipt of the predetermined signals from the inter-module communication bus (12), the interrupt mask;
second interrupt means (16) for selectively responding to an interrupt request dependent on a comparison between a priority level of the interrupt request and the interrupt mask stored in the interrupt mask means (16); and
the clock signal generation means (16) being responsive to the second interrupt means to resume providing the clock signal to the central processing unit.

5. A digital computing system (10) according to any preceding claim further comprising:

means (16) for coupling the system to an external bus, said means for coupling being responsive to receipt of the predetermined signals and interrupt mask to place the predetermined signals and the interrupt mask on the external bus.

6. A digital computing system (10) according to any preceding claim wherein the digital computing system is formed on an integrated circuit.

7. A method for executing a particular instruction in a digital computing system (10), wherein the system (10) has a central processing unit (11) and an integration module (16) which are coupled by way of an inter-module communication bus (12), the central processing unit (11) having a storage circuit (22) for storing an interrupt mask value, the method comprising the steps of:

the central processing unit (11) receiving the particular instruction;
the central processing unit (11) decoding the particular instruction; and
the central processing unit (11) initiating a write bus cycle during execution of the particular instruction;
the method being characterized by the steps of:
transferring the interrupt mask value and predetermined signals from the central processing unit (11) to the integration module (16) by way of the inter-module communication bus (12) during the write bus cycle; and
the integration module (16) halting provision of clock signals to the central processing unit (11) in response to receiving the interrupt mask value and the predetermined signals from the communication bus (12).

8. A method according to claim 7 further comprising the step of: storing the interrupt mask value in a register in the integration module (16).

9. A method according to claim 7 or 8 further comprising the steps of:

receiving an interrupt request signal having a priority level;
comparing the priority level to the interrupt mask value in order to produce a comparison result; and
selectively resuming provision of the clock signals to the central processing unit (11) dependent upon the comparison result.

10. A method according to claim 7, 8 or 9 wherein the particular instruction has a mnemonic representation of "LPSTOP".

Patentansprüche

1. Ein digitales Rechnersystem (10) umfassend einen Zwischen-Modulkommunikationsbus (12): eine zentrale Verarbeitungseinheit (11) umfassend:

   eine Registereinrichtung (22) zum Speichern einer Unterbrechungsmaske;

   eine Bussteuerungseinrichtung (23), die mit dem Zwischen-Modulkommunikationsbus (12) gekoppelt ist;

   eine erste Unterbrechungseinrichtung (20) zum selektiven Reagieren auf eine Unterbrechungsanforderung, die über die Bussteuerungseinrichtung (23) von dem Zwischen-Modulkommunikationsbus (12) in Abhändigkeit von einem Vergleich zwischen einem Prioritätsniveau .der Unterbrechungsanforderung und der Unterbrechungsmaske erhalten worden ist; und

   eine Ausführungseinrichtung (21) zum Ausführen von Befehlen; und

   ein Integrationsmodul (16), der mit dem Zwischen-Modulkommunikationsbus (12) gekoppelt ist und umfaßt:

   eine Taktsignal-Erzeugungseinrichtung (15) zum Bereitstellen eines Taktsignals für die zentrale Bearbeitungseinheit über den Zwischen-Modulkommunikationsbus (17)

   dadurch gekennzeichnet, daß

   die Ausführungseinrichtung auf einen bestimmten Befehl reagiert, indem die Bussteuerungseinrichtung (23) veranlaßt wird, vorbestimmte Signale und die Unterbrechungsmaske auf den Zwischen-Modulkommunikationsbus (12) zu geben, und

   die Taktsignal-Erzeugungseinrichtung aufhört, das Taktsignal an die zentrale Verarbeitungseinheit in Reaktion auf den Empfang der vorbestimmten Signale und der Unterbrechungsmaske von dem Zwischen-Modulkommunikationsbus (12) zu liefern.

2. Ein digitales Rechnersystem (10) gemäß Anspruch 1, daß ferner umfaßt:

   eine erste logische Einrichtung (10), die mit dem Kommunikationsbus zum Speichern der Signale gekoppelt ist, die die Unterbrechungsmaske darstellen;

   eine zweite, logische Einrichtung (16), die mit der ersten, logischen Einrichtung gekoppelt und nur aktiv ist, wenn die Taktsignal-Erzeugungslogik kein Taktsignal an die zentrale Verarbeitungseinheit zum Reagieren auf ein Unterbrechungsanforderungssignal liefert, indem ein Prioritätsniveau des Unterbrechungsanforderungssignals mit den Inhalten der ersten, logischen Einrichtung (16) verglichen wird.

   eine dritte, logische Einrichtung (16) um selektiv zu bewirken, daß die Taktsignal-Erzeugungslogik das Bereitstellen des Taktsignals für die zentrale Verarbeitungseinheit in Abhängigkeit von einem Ergebnis des Vergleichs wieder aufnimmt, der durch die zweite, logische Einrichtung ausgeführt worden ist; und

   eine Taktsignal-Erzeugungslogik, die fortlaufend ein Taktsignal an die erste, zweite und dritte, logische Einrichtung liefert.

3. Ein digitales Rechnersystem gemäß Anspruch 1 oder 2, ferner **dadurch gekennzeichnet**, daß

   die Ausführungseinrichtung (21) ferner auf den Bestimmten der Befehle reagiert, indem in der Registereinrichtung (22) ein unmittelbares Bitfeld des bestimmten der Befehle gespeichert wird.

4. Ein digitales Rechnersystem (10) gemäß Anspruch 1, das ferner umfaßt:

   eine Unterbrechungsmaskeneinrichtung (16) zum Speichern der Unterbrechungsmaske in Reaktion auf das Empfangen der vorbestimmten Signale von dem Zwischen-Modulkommunikationsbus (12);

eine zweite Unterbrechungseinrichtung (16) zum selektiven Reagieren auf eine Unterbrechungsanforderung in Abhängigkeit von einem Vergleich zwischen einem Prioritätsniveau der Unterbrechungsanforderung und der Unterbrechungsmaske, die in der Unterbrechungsmaskeneinrichtung (16) gespeichert ist; und

die Taktsignal-Erzeugungseinrichtung (16) auf die zweite Unterbrechungseinrichtung reagiert, um das Bereitstellen des Taktsignals für die zentrale Verarbeitungseinheit wieder aufzunehmen.

5. Ein digitales Rechnersystem (10) gemäß irgendeinem vorhergehenden Anspruch, ferner umfassend:

eine Einrichtung (16) zum Koppeln des Systems mit einem externen Bus, wobei die genannte Kopplungseinrichtung auf das Erhalten der vorbestimmten Signale und der Unterbrechungsmaske reagiert, um die vorbestimmten Signale und die Unterbrechungsmaske auf den externen Bus zu geben.

6. Ein digitales Rechnersystem (16) gemäß irgendeinem vorhergehenden Anspruch, wobei das digitale Rechnersystem auf einer integrierten Schaltung gebildet ist.

7. Verfahren zum Ausführen eines bestimmten Befehls in einem digitalen Rechnersystem (10), wobei das System (10) eine zentrale Verarbeitungseinheit (11) und einem Integrationsmodul (16) hat, die über einen Zwischen-Modulkommunikationsbus (12) gekoppelt sind, die zentrale Verarbeitungseinheit (11) eine Speicherschaltung (22) zum Speichern eines Unterbrechungsmaskenwertes aufweist, das Verfahren die Schritte umfaßt:

die zentrale Verarbeitungseinheit (11) erhält den bestimmten Befehl;

die zentrale Verarbeitungseinheit (11) dekodiert den bestimmten Befehl; und

die zentrale Verarbeitungseinheit (11) initiiert einen Schreib-Buszyklus während der Ausführung des bestimmten Befehls;

wobei das Verfahren durch die Schritte **gekennzeichnet ist**:

Übertragen des Unterbrechungsmaskenwertes und der vorbestimmten Signale von der zentralen Verarbeitungseinheit (11) zu dem Integrationsmodul (16) mittels des Zwischen-Modulkommunikationsbusses (12) während des Schreib-Buszyklus; und

der Integrationsmodul (16) hält das Bereitstellen von Taktsignalen für die zentrale Verarbeitungseinheit (11) in Reaktion auf das Erhalten des Unterbrechungsmaskenwertes und der vorbestimmten Signale von dem Kommunikationsbus (12) an.

8. Ein Verfahren gemäß Anspruch 7, das ferner den Schritt umfaßt:

Speichern des Unterbrechungsmaskenwertes in einem Register in dem Integrationsmodul (16).

9. Ein Verfahren gemäß Anspruch 7 oder 8, das ferner die Schritte umfaßt:

Erhalten eines Unterbrechungsanforderungssignals, das ein Prioritätsniveau hat;

Vergleichen des Prioritätsniveaus mit dem Unterbrechungsmaskenwert, um ein Vergleichsergebnis zu erzeugen; und

selektives Wiederaufnehmen der Bereitstellung von Taktsignalen für die zentrale Verarbeitungseinheit (11) in Abhändigkeit von dem Vergleichsergebnis.

10. Ein Verfahren gemäß Anspruch 7, 8 oder 9, wobei der bestimmte Befehl eine mnemonische Darstellung von "LPSTOP" ist.

Revendications

1. Système de traitement numérique (10) comprenant un bus de communications intermodules (12), une unité centrale de traitement (11) comprenant:

   - des moyens de registre (22) pour mémoriser un masque d'interruption;
   - un régisseur (ou "contrôleur") de bus (23) connecté au bus de communications intermodules (12);
   - des premiers moyens d'interruption (30) pour répondre sélectivement à une demande d'interruption reçue par l'intermédiaire du régisseur de bus (23) en provenance du bus de communications intermodules (12) selon le résultat d'une comparaison entre un niveau de priorité de la demande d'interruption et le masque d'interruption;
   - des moyens d'exécution (21) pour exécuter les instructions; et
   - un module d'intégration (16) connecté au bus de communications intermodules (12), comprenant:
   - des moyens de génération de signal d'horloge (16) pour fournir, par l'intermédiaire du bus de communications intermodules (12), un signal d'horloge à l'unité centrale de traitement,

   caractérisé en ce que:

   - les moyens d'exécution sont sensibles à une instruction précise en amenant le régisseur de bus (23) à placer des signaux prédéterminés et le masque d'interruption sur le bus de communications intermodules (12); et
   - les moyens de génération de signal d'horloge cessent de fournir le signal d'horloge à l'unité centrale de traitement en réponse à la réception des signaux prédéterminés et du masque d'interruption issus du bus de communications intermodules (12).

2. Système de traitement numérique (10) selon la revendication 1, comprenant en outre:

   - des premiers moyens de logique (16) connectés au bus de communications intermodules pour mémoriser les signaux représentant le masque d'interruption;
   - des deuxièmes moyens de logique (16) connectés aux premiers moyens de logique et étant actifs uniquement si la logique de génération de signal d'horloge n'est pas en train de fournir le signal d'horloge à l'unité centrale de traitement pour répondre à un signal de demande d'interruption en comparant un niveau de priorité du signal de demande d'interruption au contenu des premiers moyens de logique (16);
   - des troisièmes moyens de logique (16) pour amener sélectivement la logique de génération de signal d'horloge à reprendre la fourniture du signal d'horloge à l'unité centrale de traitement selon le résultat de la comparaison effectuée par les deuxièmes moyens de logique; et
   - la logique de génération de signal d'horloge fournit en permanence un signal d'horloge aux premiers, deuxièmes et troisièmes moyens de logique.

3. Système de traitement numérique selon la revendication 1 ou 2, caractérisé en outre en ce que:

   - les moyens d'exécution (21) répondent en outre à une instruction précise parmi les instructions en mémorisant dans les moyens de registre (22) un champ de binaires immédiat de la une instruction précise.

4. Système de traitement numérique (10) selon la revendication 1, comprenant en outre:

   des moyens de masque d'interruption (16) pour mémoriser, en réponse à la réception des signaux prédéterminés issus du bus de communications intermodules (12), le masque d'interruption;

   - des deuxièmes moyens d'interruption (16) pour répondre sélectivement à une demande d'interruption selon le résultat d'une comparaison entre un niveau de priorité de la demande d'interruption et le masque d'interruption mémorisé dans les moyens de masque d'interruption (16); et
   - les moyens de génération de signal d'horloge étant sensibles aux deuxièmes moyens d'interruption pour recommencer de fournir le signal d'horloge à l'unité centrale de traitement.

5. Système de traitement numérique (10) selon l'une quelconque des revendications précédentes, comprenant en outre:

   - des moyens (16) pour connecter le système à un bus extérieur, lesdits moyens pour connecter étant sensibles à la réception des signaux prédéterminés et du masque d'interruption pour placer les signaux prédéterminés et le masque d'interruption sur le bus extérieur.

6. Système de traitement numérique (10) selon l'une quelconque des revendications précédentes, dans lequel le

système de traitement numérique est formé sur un circuit intégré.

7. Procédé pour exécuter une instruction précise dans un système de traitement numérique (10), dans lequel le système (10) a une unité centrale de traitement (11) et un module d'intégration (16) qui sont connectés au moyen d'un bus de comunications intermodules (12), l'unité centrale de traitement (11) ayant un circuit de mémorisation (22) pour mémoriser une valeur de masque d'interruption, le procédé comprenant les étapes dans lesquelles:

- l'unité centrale de traitement (11) reçoit l'instruction précise;
- l'unité centrale de traitement (11) décode l'instruction précise; et
- l'unité centrale de traitement (11) déclenche un cycle d'écriture du bus pendant l'exécution de l'instruction précise,

le procédé étant caractérisé par les étapes consistant à:

- transférer la valeur de masque d'interruption et des signaux prédéterminés de l'unité centrale de traitement (11) au module d'intégration (16) au moyen du bus de communications intermodules (12) pendant le cycle d'écriture du bus; et
- le module d'intégration (16) cessant de fournir les signaux d'horloge à l'unité centrale de traitement (11) en réponse à la réception de la valeur de masque d'interruption et des signaux prédéterminés issus du bus de communications intermodules (12).

8. Procédé selon la revendication 7, comprenant en outre l'étape consistant à:

- mémoriser la valeur de masque d'interruption dans un registre du module d'intégration (16).

9. Procédé selon la revendication 7 ou 8, comprenant les étapes consistant à:

- recevoir un signal de demande d'interruption ayant un niveau de priorité;
- comparer le niveau de priorité à la valeur de masque d'interruption afin de produire un résultat de comparaison; et
- reprendre sélectivement la fourniture des signaux d'horloge à l'unité centrale de traitement (11) selon le résultat de la comparaison.

10. Procédé selon la revendication 7, 8 ou 9, dans lequel l'instruction précise a la représentation mnémonique "LPSTOP" pour décodage par ladite unité centrale de traitement (11).

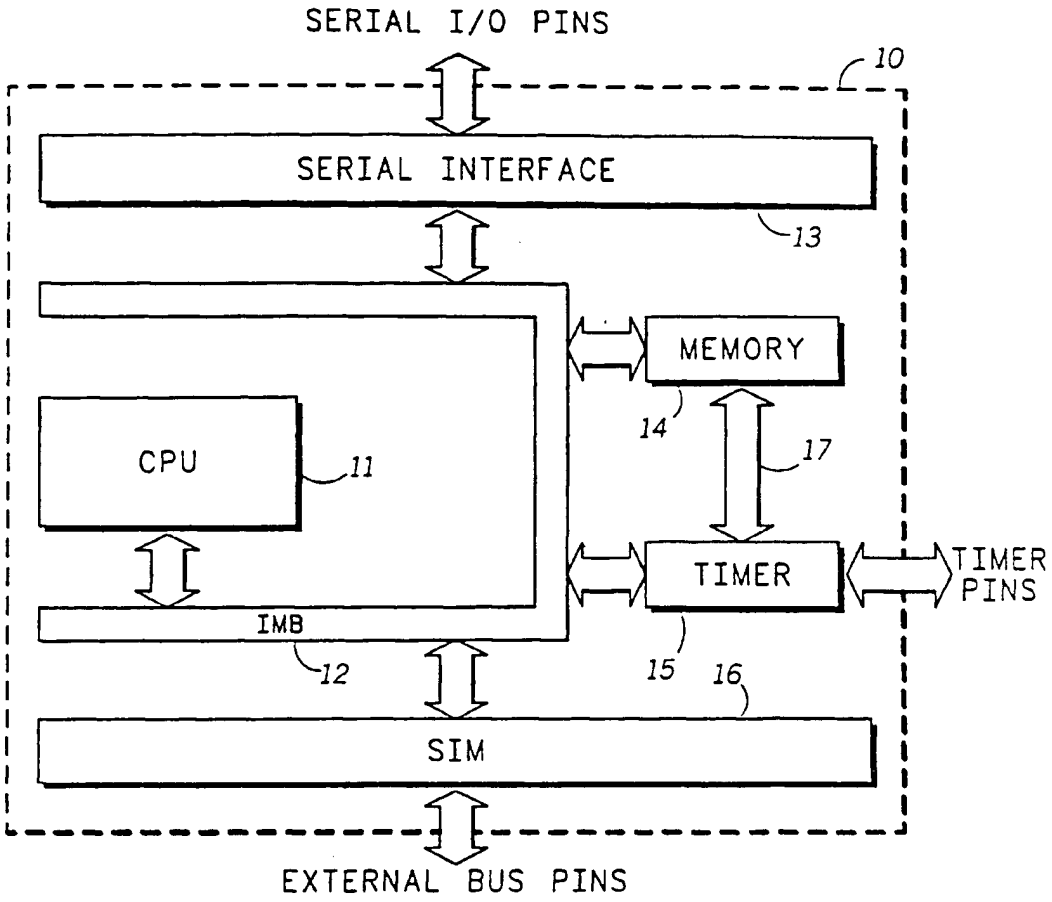
SERIAL I/O PINS
10
SERIAL INTERFACE
13
MEMORY
14
17
CPU
11
TIMER
TIMER PINS
IMB
12
15
16
SIM
EXTERNAL BUS PINS

FIG.1

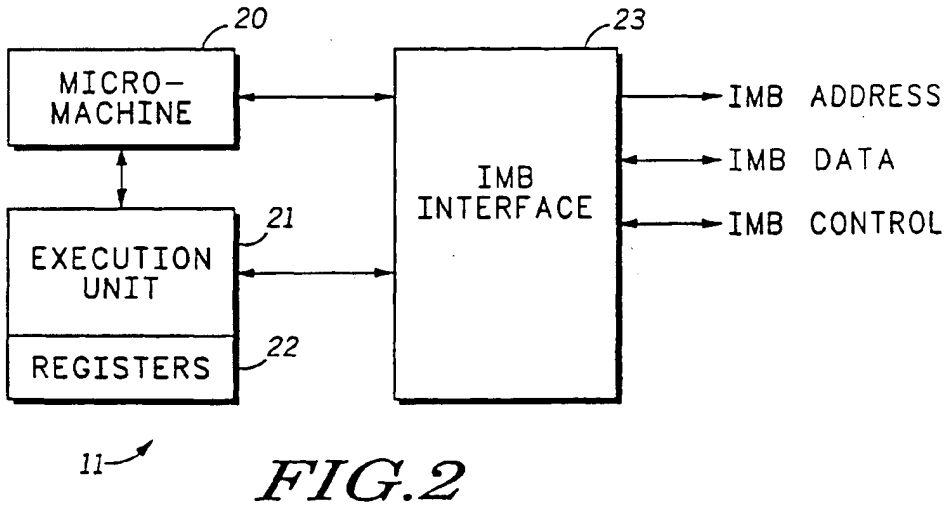
20
23
MICRO-MACHINE
IMB INTERFACE
IMB ADDRESS
IMB DATA
IMB CONTROL
EXECUTION UNIT
21
REGISTERS
22
11

FIG.2

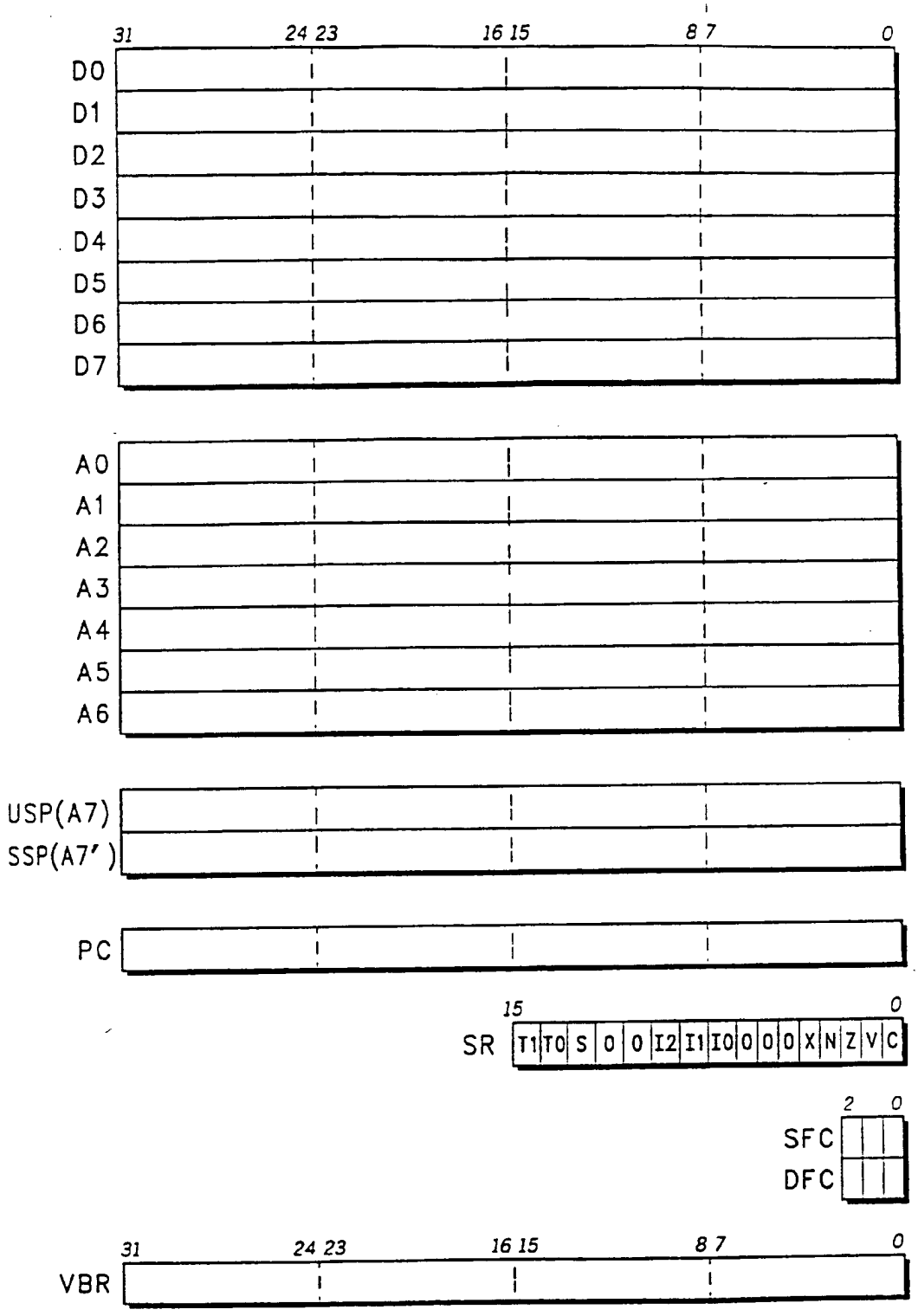
31
24 23
16 15
8 7
0
D0
D1
D2
D3
D4
D5
D6
D7
A0
A1
A2
A3
A4
A5
A6
USP(A7)
SSP(A7')
PC
15
0
SR
T1 T0 S 0 0 I2 I1 I0 0 0 0 X N Z V C
2
0
SFC
DFC
31
24 23
16 15
8 7
0
VBR

FIG.3

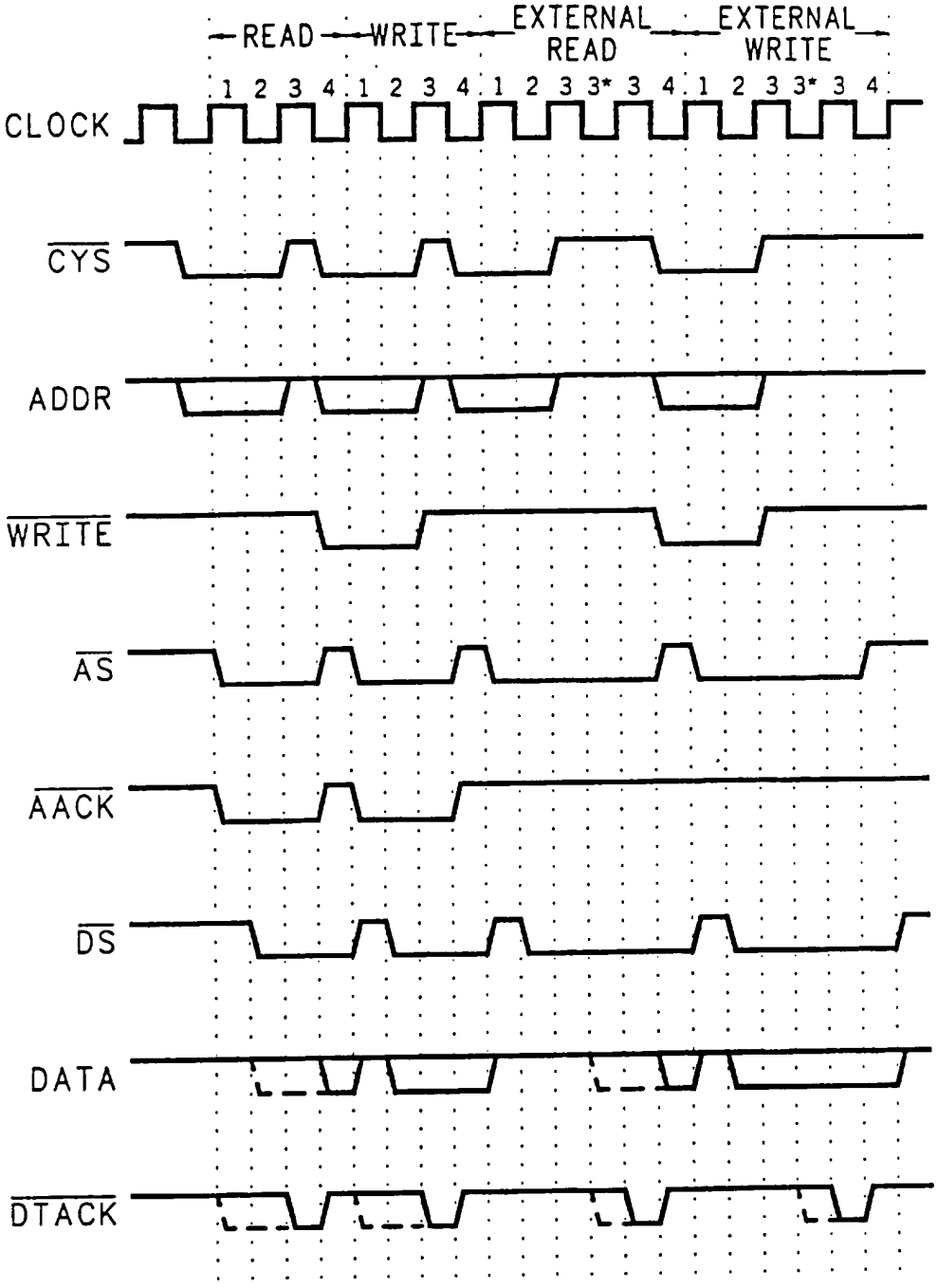
READ
WRITE
EXTERNAL READ
EXTERNAL WRITE
1 2 3 4 1 2 3 4 1 2 3 3* 3 4 1 2 3 3* 3 4
CLOCK
CYS
ADDR
WRITE
AS
AACK
DS
DATA
DTACK

FIG.4